Pascal EBNF Definition

This document is an *Extended Backus-Naur Form* definition of the Pascal programming language.

Categories of Syntactic Classes

Programs and Blocks, Procedure and Function Definitions, Statements, Expressions, Types, Record Fields, Input/Output, Variable and Identifier Categories, Low Level Definitions

Alphabetical List of Syntactic Classes

- actual-function, actual-parameter, actual-parameter-list, actual-procedure, actual-value, actual-variable, addition-operator, array-type, array-variable, assignment-statement
- base-type, block, bound-identifier, bound-specification
- case-label-list, case-limb, case-statement, component-variable, compound-statement, conditional-statement, conformant-array-schema, constant, constant-definition, constant-definition-part, constant-identifier
- declaration-part, digit, digit-sequence, directive
- element-list, element-type, entire-variable, enumerated-type, expression, expression-list
- factor, field-designator, field-identifier, field-list, field-width, file-buffer, file-component-type, file-type, file-variable, final-expression, fixed-part, for-statement, formal-parameter-list, formal-parameter-section, fraction-length, function-body, function-declaration, function-designator,function-heading, function-identification, function-identifier, function-parameter-section
- goto-statement
- identifier, identifier-list, if-statement, index-type, indexed-variable, initial-expression, integer-number
- label, label-declaration-part, letter, lower-bound
- multiplication-operator
- number
- ordinal-type-identifier, output-list, output-value
- packed-conformant-array-schema, parameter-type, pointer-type, pointer-variable, procedure-and-function-declaration-part, procedure-body, procedure-declaration, procedure-heading, procedure-identification, procedure-identifier, procedure-parameter-section, procedure-statement,program, program-heading
- real-number, record-section, record-type, record-variable, referenced-variable, relational-operator, repeat-statement, repetitive-statement, result-type
- scale-factor, set, set-type, sign, simple-expression, simple-statement, simple-type, statement, statement-part, statement-sequence, string, string-character, structured-statement, structured-type, subrange-type
- tag-field, term, type, type-definition, type-definition-part, type-identifier
- unpacked-conformant-array-schema, unpacked-structured-type, unsigned-digit-sequence, upper-bound
- value-parameter-section, variable, variable-declaration, variable-declaration-part, variable-identifier, variable-list, variable-parameter-section, variant, variant-part
- while-statement, with-statement

Definitions of Syntactic Classes

Programs and Blocks

program
    program-heading block "."
program-heading
    program identifier "(" identifier-list ")" ";"
block
    declaration-part statement-part
declaration-part
    [label-declaration-part]
    [constant-definition-part]
    [type-definition-part]
    [variable-declaration-part]
    procedure-and-function-declaration-part
label-declaration-part
    label label { "," label } ";"
constant-definition-part
    const constant-definition ";" { constant-definition ";" }
constant-definition
    identifier "=" constant
type-definition-part
    type type-definition ";" { type-definition ";" }
type-definition
    identifier "=" type
variable-declaration-part
    var variable-declaration ";" { variable-declaration ";" }
variable-declaration
    identifier-list ":" type
procedure-and-function-declaration-part
    { (procedure-declaration | function-declaration) ";" }
procedure-declaration
    procedure-heading ";" procedure-body |
    procedure-heading ";" directive |
    procedure-identification ";" procedure-body
procedure-body
    block
function-declaration
    function-heading ";" function-body |
    function-heading ";" directive |
    function-identification ";" function-body
function-body
    block
directive
    forward | compiler-defined-directives
statement-part
    begin statement-sequence **end**

Procedure and Function Definitions

procedure-heading
    procedure identifier [formal-parameter-list]
function-heading
    function identifier [formal-parameter-list] ":" result-type
result-type
    type-identifier
procedure-identification
    procedure procedure-identifier
function-identification
    function function-identifier
formal-parameter-list
    "(" formal-parameter-section { ";" formal-parameter-section } ")"
formal-parameter-section
    value-parameter-section |
    variable-parameter-section |
    procedure-parameter-section |
    function-parameter-section
value-parameter-section
    identifier-list ":" parameter-type
variable-parameter-section
    var identifier-list ":" parameter-type
procedure-parameter-section
    procedure-heading
function-parameter-section
    function-heading
parameter-type
    type-identifier | conformant-array-schema
conformant-array-schema
    packed-conformant-array-schema |
    unpacked-conformant-array-schema
packed-conformant-array-schema
    packed array "[" bound-specification "]" **of** type-idnetifier
unpacked-conformant-array-schema
    array "[" bound-specification { ";" bound-specification } "]"
    of (type-identifier | conformant-array-schema)
bound-specification
    identifier ".." identifier ":" ordinal-type-identifier
ordinal-type-identifier
    type-identifier

Statements

statement-sequence
    statement { ";" statement }
statement
    [label ":"] (simple-statement | structured-statement)
simple-statement
    [assignment-statement | procedure-statement | goto-statement]
assignment-statement
    (variable | function-identifier) ":=" expression
procedure-statement
    procedure-identifier [actual-parameter-list]
goto-statement
    goto label
structured-statement
    compound-statement | repetitive-statement | conditional-statement | with-statement
compound-statement
    begin statement-sequence **end**
repetitive-statement
    while-statement | repeat-statement | for-statement
while-statement
    while expression **do** statement
repeat-statement
    repeat statement-sequence **until** expression
for-statement
    for variable-identifier ":=" initial-expression (**to** | **downto**) final-expression **do** statement
initial-expression
    expression
final-expression
    expression
conditional-statement
    if-statement | case-statement
if-statement
    if expression **then** statement [**else** statement]
case-statement
    case expression **of**
    case-limb { ";" case-limb } [";"]
    end
case-limb
    case-label-list ":" statement
case-label-list
    constant { "," constant }
with-statement
    with record-variable { "," record-variable } **do** statement
actual-parameter-list
    "(" actual-parameter { "," actual-parameter } ")"
actual-parameter
    actual-value | actual-variable | actual-procedure | actual-function

actual-value
    expression
actual-procedure
    procedure-identifier
actual-function
    function-identifier

Expressions

expression
    simple-expression [relational-operator simple-expression]
simple-expression
    [sign] term { addition-operator term }
term
    factor { multiplication-operator factor }
factor
    variable | number | string | set | **nil** | constant-identifier | bound-identifier | function-designator | "(" expression ")" | **not** factor
relational-operator
    "=" | "<>" | "<" | "<=" | ">" | ">=" | "in"
addition-operator
    "+" | "-" | **or**
multiplication-operator
    "*" | "/" | **div** | **mod** | **and**
variable
    entire-variable | component-variable | referenced-variable
entire-variable
    variable-identifier | field-identifier
component-variable
    indexed-variable | field-designator | file-buffer
indexed-variable
    array-variable "[" expression-list "]"
field-designator
    record-variable "." field-identifier
set
    "[" element-list "]"
element-list
    [expression { "," expression }]
function-designator
    function-identifier [actual-parameter-list]
file-buffer
    file-variable "^"

Types

type
    simple-type | structured-type | pointer-type | type-identifier
simple-type
    subrange-type | enumerated-type
enumerated-type
    "(" identifier-list ")"
subrange-type
    lower-bound ".." upper-bound
lower-bound
    constant
upper-bound
    constant
structured-type
    [**packed**] unpacked-structured-type
unpacked-structured-type
    array-type | record-type | set-type | file-type
array-type
    array "[" index-type { "," index-type } "]" **of** element-type
index-type
    simple-type
element-type
    type
record-type
    record field-list **end**
set-type
    set of base-type
base-type
    type
file-type
    file of file-component-type
file-component-type
    type
pointer-type
    "^" type-identifier

Record Fields

field-list
    [(fixed-part [";" variant-part] | variant-part) [";"]]
fixed-part
    record-section { ";" record-section }
record-section
    identifier-list ":" type
variant-part
    case tag-field type-identifier **of** variant { ";" variant }

tag-field
    [identifier ":"]
variant
    case-label-list ":" "(" field-list ")"

Input/Output

output-list
    output-value { "," output-value }
output-value
    expression [";" field-width [":" fraction-length]]
field-width
    expression
fraction-length
    expression

Variable and Identifier Categories

identifier
    letter { letter | digit }
file-variable
    variable
referenced-variable
    pointer-variable "^"
record-variable
    variable
pointer-variable
    variable
actual-variable
    variable
array-variable
    variable
field-identifier
    identifier
constant-identifier
    identifier
variable-identifier
    identifier
type-identifier
    identifier
procedure-identifier
    identifier
function-identifier
    identifier
bound-identifier
    identifier

Low Level Definitions

variable-list
    variable { "," variable }
identifier-list
    identifier { "," identifier }
expression-list
    expression { "," expression }
number
    integer-number | real-number
integer-number
    digit-sequence
real-number
    digit-sequence "." [digit-sequence] [scale-factor] |
    digit-sequence scale-factor
scale-factor
    ("E" | "e") [sign] digit-sequence
unsigned-digit-sequence
    digit { digit }
digit-sequence
    [sign] unsigned-digit-sequence
sign
    "+" | "-"
letter
    "A" | "B" | "C" | "D" | "E" | "F" | "G" | "H" | "I" | "J" | "K" | "L" | "M" | "N" | "O"
    | "P" | "Q" | "R" | "S" | "T" | "U" | "V" | "W" | "X" | "Y" | "Z" | "a" | "b" | "c" | "d"
    | "e" | "f" | "g" | "h" | "i" | "j" | "k" | "l" | "m" | "n" | "o" | "p" | "q" | "r" | "s" | "t" |
    "u" | "v" | "w" | "x" | "y" | "z"
digit
    "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9"
string
    """ string-character { string-character } """
string-character
    any-character-except-quote | """
label
    integer-number
constant
    [sign] (constant-identifier | number) | string